

Appl: No. 09/760,405  
Amdt. dated 4/25/05  
Reply to Office action of 2/23/05

REMARKS/ARGUMENTS

Reconsideration of the application is requested.

Claims 1 and 4-12 remain in the application. Claims 2 and 3 were previously canceled.

In item 4 on page 2 of the above-identified Office Action, claims 1-9 and 11 have been rejected as being unpatentable over Suzuki (US 6,499,096) in view of Kahle et al. (US 5,913,925) (hereinafter "Kahle") under 35 U.S.C. § 103(a).

Applicant notes that only claims 1 and 4-12 are present in the subject application. Further, in the body of the rejection the Examiner refers to claims 1 and 4-11. However, in the statement of the rejection the Examiner includes claims 2 and 3 which have been canceled and does not mention claim 10. Therefore, it is presumed that the Examiner intended to reject claims 1 and 4-12, and applicant will address the rejection on the basis that claims 1 and 4-12 were rejected.

As will be explained below, it is believed that the claims were patentable over the cited art in their current form and, therefore, the claims have not been amended to overcome the references.

Before discussing the prior art in detail, it is believed that a brief review of the invention as claimed, would be helpful.

Claim 1 calls for, *inter alia*, a data-processing device **for processing in parallel a plurality of independent processes**, having:

a program memory having stored therein at least one compiled program **with a multiplicity N of independent processes**, the compiled program including information on parallelism and **including a multiplicity of bundles with a plurality of instructions of a process, the instructions of a bundle being executable in parallel**;

a branching control unit connected to and addressing said program memory;

a register for storing flags and data which are switched in dependence on a process being executed;

and controlling an output of instructions to be processed in parallel in dependence on information contained in the instructions and included in a compiling time of the program;

a number N of instruction buffers being connected in parallel downstream of said program memory for storing instructions

read out from said program memory, an instruction bundle being read into one of said instruction buffers and a second instruction bundle associated with a different process being read into another one of said instruction buffers; and

an instruction output selector being connected to and controlled by said process flow control unit **causing said instruction output selector to read out instructions from said instruction buffers and output N instructions in parallel,**

**said instruction output selector having a multiplexer logic and selecting one of either one instruction from a first instruction buffer and one instruction from a second instruction buffer, or two instructions from one of said first and second instruction buffers.** (emphasis added)

A major advantage of the present claimed invention is that not only the independent processes can be executed in parallel but also instructions within one bundle can be executed in parallel. This advantage can be further explained with reference to Fig. 1 of the instant application. A first bundle of a first process is loaded into the instruction buffer (13) and a second bundle of a second process is loaded into the second instruction buffer (14). It is known that all instructions of the first bundle are independent with regard

to the instructions of the second bundle, because the first bundle belongs to the first process and the second bundle belongs to the second process and both processes are independent, as recited in claim 1. In one case, the instruction issue selector (15) selects from each of the two instruction buffers (13, 14) one instruction, so these instructions can be fed to the execution units (19, 20) and thus be processed in parallel. In this scenario both the first and the second processes are executed in parallel and both execution units (19, 20) are under full load. If an instruction of the first process requests a memory access this instruction cannot be executed immediately by the execution unit (19). Additionally, further instructions of the first process may be not executable in parallel to the previously mentioned instruction. Thus, the execution unit (19) would be without load. An advantage of the present invention is that in case a bundle of two or more instructions is present in the second instruction buffer (14), two of these instructions could be loaded by the instruction issue selector (15) and subsequently executed in parallel by both execution units (19,20). Thus, some load for the execution unit (19) may be maintained even if no instructions of the first process can be executed at the moment.

The reference numbers used above are for explanation purposes only and are not intended to be limiting in anyway of the claimed invention.

Suzuki discloses a VLIW (very long instruction word processor) executing a compressed program code. The compressed code has left sub-instructions to be executed by a left process unit 6 and right sub-instructions to be executed by a right processing unit 7. In Suzuki an exchanging portion 8, 8' is able to identify sub-instructions as left sub-instructions and right sub-instructions, respectively. In one step a first instruction and a second instruction are loaded from a right and a left part of a memory 15 by the exchanging portion 8'. If the first and the second instruction are a left and a right sub-instruction, respectively, they are passed in parallel to the respective processing units. When both instructions are left sub-instructions a first of these left sub-instructions is passed immediately to the left processing unit and the second sub-instruction is switched by the exchanging portion 8' from the path having the right processing unit to the path having the left processing unit and executed afterwards. Thus, these two left sub-instructions are executed, selected, and output sequentially by the exchanging portion and not in parallel. There is no suggestion or teaching of doing this in parallel as recited in the instant claims.

In contrast, the present invention selects in one mode two instructions from one of the two instruction buffers and outputs the two instructions in parallel as recited in the last paragraph of claim 1.

Nor does Suzuki disclose different independent processes stored within a program memory as set forth in claim 1. The Examiner acknowledges that Suzuki is deficient and attempts to makeup for the deficiencies by combining features of Kahle with Suzuki. Applicant submits that such a combination is not obvious as alleged by the Examiner, nor is there sufficient basis in the primary Suzuki reference to warrant modifying it by Kahle as proposed by the Examiner. Merely stating that such a combination would have been obvious "to improve process performance" as alleged by the Examiner is wishful thinking by the Examiner and not sufficient reason or motivation to cause one skilled in the art to modify Suzuki as proposed. The only apparent reason for such a combination is hindsight reconstruction of the prior art in view of applicant's invention.

Moreover, a simple use of a multi-scalar device would not improve the execution of a plurality of independent processes as it does in the present claimed invention. General multi-scalar devices only execute one independent process and in a

reduced load situation would nor execute instructions of a different independent process. This is primarily because the instructions are to be executed sequentially. The present invention overcomes this problem by providing additional information within each independent process forming bundles of instructions which may be executed in parallel. Upon a lower load of the processing unit such a bundle may be split and one instruction of this bundle may be executed by the processing unit under a low load. Thus, such a device using independent processes is not taught or suggested by Suzuki in view of Kahle.

The references do not show a "a program memory having stored therein at least one compiled program with a multiplicity N of independent processes" and "an instruction output selector being connected to and controlled by said process flow control unit causing said instruction output selector to read out instructions from said instruction buffers and output N instructions in parallel, said instruction output selector having a multiplexer logic and selecting one of either one instruction from a first instruction buffer and one instruction from a second instruction buffer, or two instructions from one of said first and second instruction buffers" as recited in claim 1 of the instant application.

In item 16 on page 8 of the above-identified Office Action, claim 12 has been rejected as being unpatentable over Suzuki in view of Kahle and further in view of Allen et al. (US 6,404,752) (hereinafter "Allen") under 35 U.S.C. § 103(a).

Applicant notes that in the rejection the Examiner refers on page 9 of the subject Office Action to the Gupta reference. However, the rejection of claim 12 was based on Suzuki in view of Kahle and Allen. Gupta was not relied on by the Examiner. Therefore, it is presumed that the reference to Gupta is superfluous and was not intended to be included in the rejection of claim 12. Therefore, applicant will not discuss Gupta in this reply, although the discussion of the Gupta reference on pages 13-16 of the previously-filed September 30, 2004 response in the subject application is equally applicable in this instance if the Examiner intended to include Gupta in the rejection.

The foregoing discussion of Suzuki and Kahle are equally applicable in the rejection.

Allen does not overcome the deficiencies of Suzuki or Kahle, individually or in combination, as discussed above. Allen discloses a network switch in which data flow handling and flexibility is enhanced by a network processor capable of

cooperating with other elements, such as an optional switching fabric device, to execute instructions that direct the flow of data within a network. The network processor includes a plurality of cooperating interface processors and various peripheral elements formed on a semiconductor substrate. Allen also discloses, "network processors can increase bandwidth and solve latency problems in a broad range of application by allowing networking tasks previously handled in software to be executed in hardware" (col. 2, lines 64-67).

Clearly, the references do not show "a program memory having stored therein at least one compiled program with a multiplicity N of independent processes" and "an instruction output selector being connected to and controlled by said process flow control unit causing said instruction output selector to read out instructions from said instruction buffers and output N instructions in parallel, said instruction output selector having a multiplexer logic and selecting one of either one instruction from a first instruction buffer and one instruction from a second instruction buffer, or two instructions from one of said first and second instruction buffers" as recited in independent claim 1 of the instant application. Nor does Allen show or suggest "a network processor for processing layer 1 to 7 of protocol stacks in applications including LAN, ATM switches,

Appl. No. 09/760,405  
Amdt. dated 4/25/05  
Reply to Office action of 2/23/05

IP routers, and frame relays based on a system selected from the group consisting of DSL, Ethernet, and cable modems" as recited in claim 12 of the instant application.

It is accordingly believed to be clear that none of the references, whether taken alone or in any combination, either show or suggest the features of claim 1. Claim 1 is, therefore, believed to be patentable over the art. The dependent claims are believed to be patentable as well because they all are ultimately dependent on claim 1.

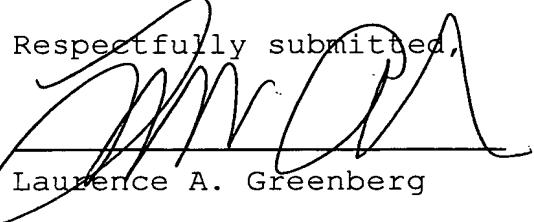
In view of the foregoing, reconsideration and allowance of claims 1 and 4-12 are solicited.

In the event the Examiner should still find any of the claims to be unpatentable, counsel would appreciate receiving a telephone call so that, if possible, patentable language can be worked out.

If an extension of time for this paper is required, petition for extension is herewith made.

Appl. No. 09/760,405  
Amdt. dated 4/25/05  
Reply to Office action of 2/23/05

Please charge any other fees that might be due with respect to Sections 1.16 and 1.17 to the Deposit Account of Lerner and Greenberg, P.A., No. 12-1099.

Respectfully submitted,  
  
Laurence A. Greenberg

**LAURENCE A. GREENBERG**  
REG. NO. 29,308

LAG/FDP/bb

April 25, 2005

Lerner and Greenberg, P.A.  
Post Office Box 2480  
Hollywood, FL 33022-2480  
Tel: (954) 925-1100  
Fax: (954) 925-1101